

Chapter 3: Getting started

Before using the VT100 Gateway

Before you can use the VT100 Gateway, you must complete these tasks:

- If necessary, install a multiport adapter board and expander box.
- Create the action and screen templates necessary to navigate through the host application and return the desired information (see Chapter 2).
- Create the `screen.conf`, `trs.conf`, `vt100.ctl`, and `com.conf` files (described in this chapter).
- Before starting IVR 2.0/I, **cd** to the `/u/ivr/exe` directory on the application processor and execute the `./trs -c` command. This will verify that the VT100 Gateway has been installed and configured correctly, and that the screen and action templates have the appropriate syntax.
- Start IVR 2.0/I on the application processor. The TRS process is automatically started when you issue the **MIVR start** command.

If you are going to connect to the host computer via modem, you will have to configure the modem with the correct settings, attach the modem to the communication port, and dial up the host computer. Also make sure that **DIAL_UP** is specified in the `com.conf` file (discussed later).

During startup, the `trs.log` file remains empty unless the verbose option (`-v`) is added. To debug the application put the TRS process into debug mode. Check the `trs.log` file or the `trs.tmp` file for errors. To put the TRS process into verbose mode go to `/u/ivr/startup` and change the `./trs -b` line so that it reads `./trs -b -v`.

screen.conf file

Since the VT100 protocol is character based, it has no built in mechanism for notifying the TRS that host output has ended and the application is ready for input. Creating the **screen.conf** file allows the TRS to quickly process host output by eliminating the time it spends waiting after host output ends.

The TRS uses the **screen.conf** file to determine the end of host transmission. The TRS checks every screen sent from the host against the screens you define in **screen.conf**. Each time the TRS recognizes a complete host screen, it allows the transaction to continue.

The TRS searches each host screen for one of the keywords in **screen.conf**. When it finds the keyword on the host screen, it then searches for the end of the screen. When it encounters the end of the screen, host output has ended so the TRS allows the transaction to continue. As an added checking mechanism, the TRS can also check for the beginning of the screen to ensure that it is passing along the correct screen.

Screen.conf must reside in the /u/ivr/3270 directory.

Screen.conf should contain the definitions of every screen the TRS will encounter - one definition per line. The order of the lines does not matter since the TRS checks every screen against all the definitions in **screen.conf**. Figure 3-1 shows the syntax of the **screen.conf** file.

Figure 3-1
screen.conf file syntax

Keyword:Begin string:End string:

The colons (:) are used as field delimiters and must be placed in the specified positions without any additional white space.

Keyword:

The keyword identifies the screen. If the keyword contains a space or a colon, then enclose the keyword in quotation marks. Since the keyword's purpose is to identify each screen, choose a unique keyword for each different screen.

Begin string: (optional)

The TRS uses this string to determine the beginning of the screen. The begin string should be the first character or string on the screen. If it contains a space or a colon, then place the string in quotation marks. If you do not want the **screen.conf** file to check for the beginning of the screen, then enter a hyphen in this field.

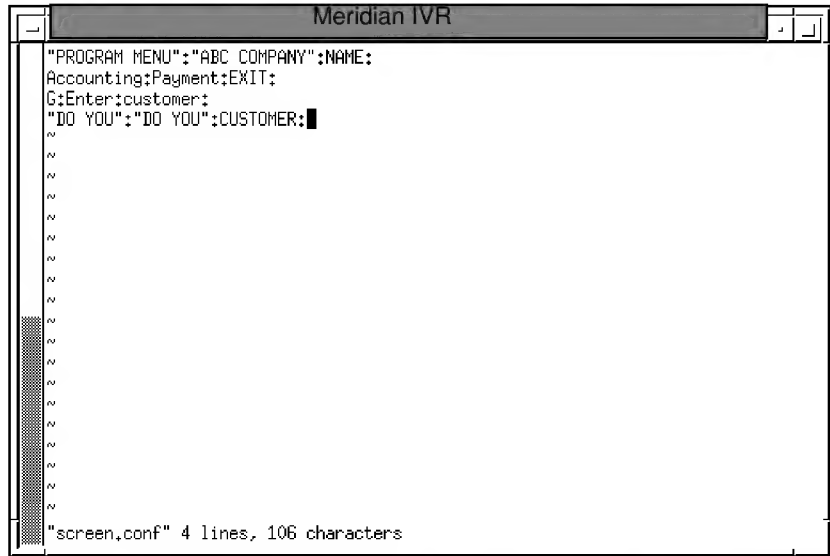
End string:

The TRS uses the end string to determine the end of the screen. The end string should be the last character or string on the screen. If the end string contains a space or a colon, then place the end string in quotation marks. You must provide an end string for the screen.conf file to be valid.

Note: The TRS considers the begin string and the end string part of the data.

See Figure 3-2 for an example of the **screen.conf** file.

Figure 3-2
screen.conf file



In Figure 3-2, **PROGRAM MENU** is the keyword for the first screen. Since it contains a “s13is” case, it also searches for the begin string **PROGRAM MENU**. Upon finding the begin and end string, the TRS continues the transaction.

If your host application screens do not have enough unique keywords or if the application demonstrates unpredictable screen behavior, then you have difficulty making a **screen.conf** file. In this case refer to the **MAX_WAIT_TIME** entry in the **com.conf** file discussed in Chapter 3.

Setting up the trs.conf file

The **trs.conf** file specifies the initial-action template for each session you define on the IVR 2.0/I application processor. The **trs.conf** file must reside in the /u/ivr/3270 directory. Figure 3-3 shows the syntax for the **trs.conf** file.

Figure 3-3
trs.conf file syntax

```
app-name:board-number>session-number:initial-template:heartbeat:protocol
```

The colons (:) and the greater than (>) symbols are used as field delimiters and must be placed in the specified positions without any additional white space.

app-name

The *app-name* entry identifies the application on the mainframe to which you assign the session. The name entered here is the same name you enter in the action templates that are executed by the IVR 2.0/I application.

board-number

The *board-number* entry must be 0. This entry is provided to support future development.

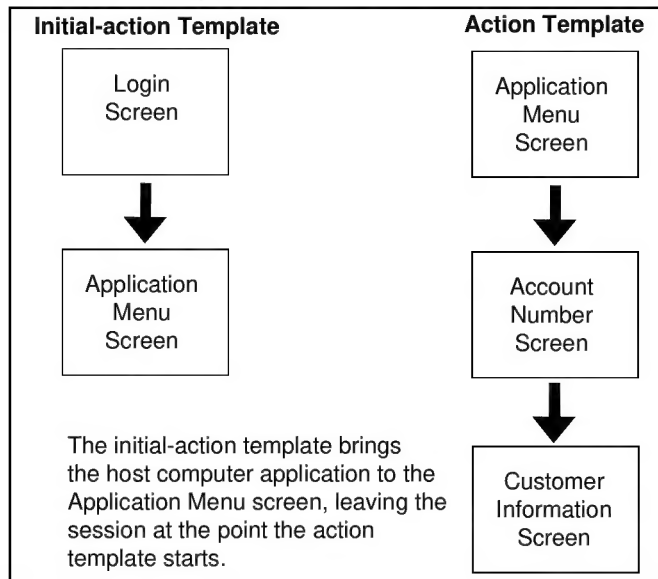
session-number

In the VT100 option, the *session-number* entry determines the number of sessions per board with session numbers ranging from 2 to 16. By using two digital boards there can be up to 15 sessions, if there is only one ACCESS link.

initial-template

The **initial-template** entry identifies an initial-action template (without the .act file extension) for setting the startup action for the specified sessions when connecting to the host computer. The start-up action brings the specified sessions to the screen on the host computer where the action templates start when processing requests. Figure 3-4 shows the flow of a sample initial-action template.

Figure 3-4
Initial-action template



The initial-action template follows the same format described earlier for action templates. The screen templates specified by the initial-action template must also be created. The screen template for a login screen (or the login prompt) must specify the correct login ID and password (if appropriate) to access the host computer. You should not specify a reset-action template in the initial-action template, although you should specify a logout-action template.

heartbeat

You can specify an optional *heartbeat* action for the application specified by *app-name*. You can use this feature to send an indication to the host that a session is still active. Some hosts log out sessions that remain idle for a period of time. You can also use the heartbeat action to check connectivity, verifying that the session remains on the appropriate screen. Typically, the *heartbeat* action contains a logout-action template which brings the session back to the login screen if the connectivity with the mainframe fails. If you do not need a heartbeat action, enter a hyphen for this action.

You specify the *heartbeat* in this format: *actiontemplate@n*, where *n* is the number of seconds between each execution of the action template when the session is idle.

The heartbeat action template uses the same syntax as the action templates described earlier in this chapter, and usually only specifies a single screen template. That screen template is usually the last screen specified in the initial-action template. Typically, your screen template would only include a *key-descriptor* line, usually the **ENTER** key and *validation-tag*.

protocol

The protocol entry indicates the communications protocol being used by *app-name*. This field is required and should be set to vt100.

Example “trs.conf” File

Consider the following example. Initial-action template files login.act and signin.act have been defined and reside in the /u/ivr/3270 directory. There are four applications which you want to access on the host computer:

- **accounting**, accesses the accounting software
- **market**, tracks stock market activity
- **banking**, retrieves credit balances
- **airline**, for purchasing tickets on a local commuter carrier

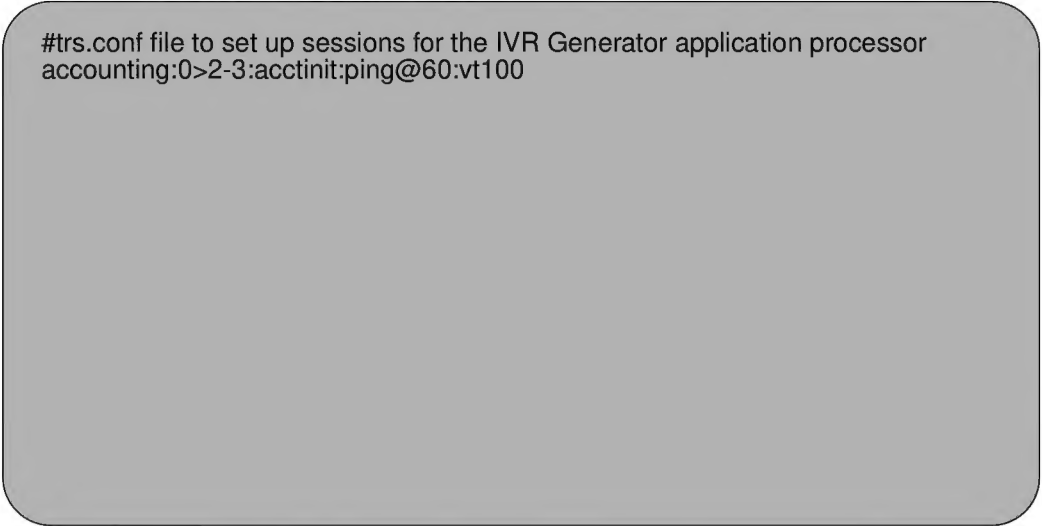
The application names shown here are not necessarily the actual names assigned on the host computer, but they are the names assigned on the application processor for the trs.conf file and all action templates that use these applications.

You want to set up the VT100 Gateway to initialize the sessions as follows:

- initialize sessions 2-3 for **accounting** with **acctinit.act**,
- initialize sessions 4-8 for **market** with **login.act**,
- initialize sessions 9-10 for **banking** with **login.act**, and
- initialize sessions 15-17 for **airline** with **signin.act**

Figure 3-5 illustrates how you should set up **trs.conf** to implement the configuration for only the accounting application.

Figure 3-5
trs.conf file for accounting application



```
#trs.conf file to set up sessions for the IVR Generator application processor  
accounting:0>2-3:acctinit:ping@60:vt100
```

The trs.conf file as shown in Figure 3-5 relates to the initial-action template shown in Figure 3-9.

Setting up the vt100.ctl file

The vt100.ctl file must be created and stored in the /u/ivr/vt100 directory before you can use the VT100 Gateway. The vt100.ctl file configures the terminal type for each session and corresponds to the sessions defined in the trs.conf file.

Each line in the vt100.ctl file uses the syntax shown in Figure 3-6.

Figure 3-6
vt100.ctl file syntax

```
session-number  device-name  terminal-type
```

Figure 3-7 shows a **vt100.ctl** file, based on the example **trs.conf** file in Figure 3-5.

Figure 3-7
vt100.ctl file for accounting application

```
2 /dev/tty1b vt100
3 /dev/tty1c vt100
```

The follow sections describe each entry in the vt100.ctl file.

session-number

The *session-number* entry specifies the session this line defines. For example, the first line in Figure 3-7 defines the terminal type to be used for session 2. The entry must be a single number, ranging from 2 to 16. You do not need to list the sessions in numerical order, and you can skip session numbers.

device-name

The *device-name* entry is the exact name of the device file in the **/dev** directory to use for the VT100 Gateway for the specified sessions. Use the full directory path (as shown in Figure 3-7) when entering the device file name. You can set all necessary operating modes (e.g., baud rate, representation parameters, flow control, etc.) in the com.conf file discussed in the next section.

terminal-type

The *terminal-type* entry defines the type of terminal to be emulated by the TRS process. For this release of the VT100 Gateway, this entry is always "vt100." This field is included to support future releases that may allow additional terminal types.

Setting up the com.conf file

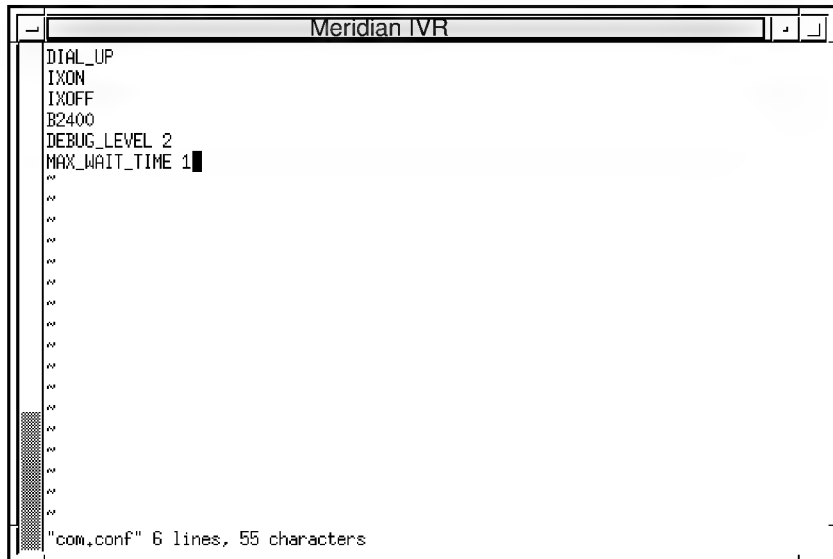
com.conf sets the communication attributes of the serial port. The **com.conf** file allows the adjustment of the terminal I/O options for the host's input device, which in this case is the IVR 2.0/I TRS process. If the **com.conf** file does not exist, the TRS uses its hardcoded default values. Table 3-1 shows the TRS default communication settings.

Table 3-1
TRS default communication settings

Baud Rate:	9600
Parity:	None
Stop Bit:	1

The **com.conf** file must be created if the baud rate is anything other than 9600. The **com.conf** file is located in the vt100 directory and a sample is shown in Figure 3-8.

Figure 3-8
com.conf file



Creating the **com.conf** file allows you to adjust the following values.

IXON

IXON allows you to stop host output by sending ASCII DC3, and restart host output by sending ASCII DC1. **IXON** prevents input queue overflow.

IXOFF

IXOFF requests that the host send start/stop characters when the input queue is nearly empty or full. **IXOFF** prevents host input queue overflow.

IXON and **IXOFF** prevent queue overflow. If both are set and the host or terminal sends more information than their respective queues can handle, the communication port stops the data flow. When the queue gets down to a manageable level, the communication port resumes the flow of information.

Baud Rate

300 to 19200. Setting the baud rate tells the TRS the rate at which it is to communicate with the host. Set the baud rate by typing **B** then the rate. For example **B4800** or **B19200**.

Parity

PAREN enables parity generation and detection.

PAREVN sets parity to even.

PARODD sets parity to odd.

CSTOPB

CSTOPB select 1 stop bit per character. If **CSTOPB** is not specified, then there are 2 stop bits per character.

DIAL_UP

Enter **DIAL_UP** in the **com.conf** file if IVR 2.0/I is going to access the host computer via modem.

DEBUG_LEVEL

1 to 5. Setting the debug level determines how much debugging information is sent to the **/u/ivr/vt100/vt100log** file.

MAXIMUM_WAIT_TIME

This parameter tells the TRS how long to wait for the output from the host to end. Only use this parameter if you can not use a **screen.conf** file. The wait time is specified in seconds.

A complete sample transaction

This section summarizes how the sample accounting application would start up, retrieve a customer's balance and reset for the next transaction. The sample transaction uses these templates in the following way:

- The initial-action template logs on to the host computer and starts up the “acct” application.
- The action template chooses the “Accounts Receivable” option from the application's menu and retrieves a customer's balance.
- The reset-action template returns to the application's menu screen and waits for the next transaction.
- The logout-action template is included in case an error occurs at any time during the transaction.

Initial-action template

Once the TRS process is invoked, the initial-action template `acctinit.act` brings the session assigned to the accounting application to the menu screen. Figure 3-9 shows the initial-action template, its supporting screen templates and the corresponding screens on the host computer.

Figure 3-9
Initial-action template for accounting application

Templates	Corresponding Screens
<pre>#initial action template to start #the accounting application #filename: acctinit.act acctinit accounting - clrinit acctlog1.scn acctlog2.scn atactmenu</pre>	<pre>login:</pre>
<pre>#screen template that logs #into the host #filename: acctlog1.scn acctlog1 1,1 login: 0,0 - vad 0,0 > ENTER 0,0 - quality 0,0 > ENTER 0,0 @ 3</pre>	<pre>login: vad Password: ***** login: vad Password: ***** Last login: Fri Oct 7 16:01:34 from Publisher ULTRIX V4.2 (Rev 96) System #9: Wed Oct 12 10:08:24 EDT 1994 \$</pre>
<pre>#screen template that starts #the accounting application #filename: acctlog2 acctlog2 0,0 ULTRIX 0,0 - acct 0,0 > ENTER 0,0 @ 5</pre>	<pre>login: vad Password: ***** Last login: Fri Oct 7 16:01:34 from Publisher ULTRIX V4.2 (Rev 96) System #9: Wed Oct 12 10:08:24 EDT 1994 \$ acct</pre>
<pre>#screen template that validates #the accounting menu screen #filename: atactmenu.scn atactmenu 1,20 ACME Accounting</pre>	<pre>ACME Accounting 1. Accounts Receivable 2. Accounts Payable 3. Reports 4. Inventory 5. Exit Enter menu selection:1</pre>

The initial-action template starts the accounting session with the host computer by accessing **accounting** from the **trs.conf** file. It then executes the screen templates **acctlog1**, **acctlog2**, then **atactmenu**.

The screen template **acctlog1.scn** does the following:

- enters the login name
- enters the password
- waits three seconds for the host to process the login action

The screen template **acctlog2.scn** does the following:

- at the system prompt, enters the command to run the application
- types the ENTER key to start the application
- waits five seconds for the host to run the accounting application

The screen template **atacctmenu.scn** validates the accounting main menu screen, but performs no action.

The **acctlog1** and **acctlog2** screen templates use “0,0” as the *row,column* location and a hyphen for the *field-tag* to indicate that all text will be entered at the current cursor position. This method is used because the login prompt and the first prompt may not always appear in the same location on the screen. Prompts vary from system to system, and can include longer, customized names.

Note: No *key-descriptor* is specified for the “atacctmenu” screen template. This means once the screen is validated, this screen remains active until the next transaction is executed.

The initial-action template shown in Figure 3-10 specifies the **clinit.act** logout-action template, and does not specify a reset-action template. Figure 3-10 shows the **clinit** logout-action template, its corresponding screen templates, and the associated application screens.

Figure 3-10
Logout-action template used by the initial-action template for accounting application

Templates	Corresponding Screens
<pre>#logout-action template for the #acctinit initial-action template #filename: clinit.act clinit accounting - - clrmenu logout</pre>	<pre>ACME Accounting 1 Accounts Receivable 2 Accounts Payable 3 Reports 4 Inventory 5 Exit Enter menu selection:</pre>
<pre>#screen template to exit the #accounting application #filename: clrmnu.scn clrmnu 1,20 ACME Accounting 0,0 - 5 0,0 > ENTER 0,0 - Y 0,0 > ENTER</pre>	<pre>ACME Accounting 1 Accounts Receivable 2 Accounts Payable 3 Reports 4 Inventory 5 Exit Enter Menu selection 5 Do you really want to exit? Y</pre>
<pre>#screen template to log off the host #computer, preparing for the initial #action template to be executed #filename: logout_scn logout 0,0 - 0,0 - logout 0,0 > ENTER</pre>	<pre>\$ logout login</pre>

The final screen template listed, **logout**, brings the host application back to the initial screen, leaving the connection open and waiting for the TRS process to login. TRS always executes the initial-action template for a session (after a pause of 30 seconds) after any logout-action template is executed.

Action template performing a transaction

As shown in Figure 3-10, the initial-action template brings the session to the application's menu screen. The action template created to get a customer's balance starts at that screen. Figure 3-11 shows the action template, its screen templates, and the corresponding host computer screens that perform the following functions:

- choose the Accounts Receivable option from the application's menu
- enter (on the line that pops up at the bottom of the screen) the customer account number from an input buffer provided by the COMI cell
- when the customer information screen displays, the template places the current balance in an output buffer to be transmitted to the COMO cell
- execute a reset-action template to return the session application to the application's menu screen

Figure 3-11
Action template for accounting application

Templates	Corresponding Screens
<pre>#action template to perform steps #required to retrieve customer's balance #filename: getbalance.act getbalance accounting clr_cust logout_cust accrec #choose Ac. Rec menu acctno #enters account number customer #retrieves balance</pre>	<pre>ACME Accounting 1 Accounts Receivable 2 Accounts Payable 3 Reports 4 Inventory 5 Exit Enter menu selection: █</pre>
<pre>#screen template to choose accounts #receivable option, filename: accrec.scn accrec 1,20 ACME Accounting 0,0 — 1 0,0 > ENTER</pre>	<pre>ACME Accounting 1 Accounts Receivable 2 Accounts Payable 3 Reports 4 Inventory 5 Exit Enter menu selection: 1 Enter account number: █</pre>
<pre>#screen template to enter acct number #in popup field, filename: acctno.scn acctno 1,11 Enter account number: 0,0 — * 0,0 > ENTER</pre>	<pre>ACME Accounting 1 Accounts Receivable 2 Accounts Payable 3 Reports 4 Inventory 5 Exit Enter menu selection: 1 Enter account number: 845-23-87</pre>
<pre>#screen template to obtain balance #filename: customer.scn customer 1,1 Account Number: 2,48 — \$9</pre>	<pre>Account Number: 845-23-87 Customer: Jane K. Smith Current Balance: 2486.14 Address: 19 Alpha Road Payment Due: 150.00 Chelmsford Payment Due Date: 4/30/93 MA 01824 Options: 1 Print invoice 2 Enter payment 3 Enter purchase 4 Exit Enter menu selection: █</pre>

The number 1 and the account number are entered at the current cursor location. Therefore, you do not need to specify a location or a **field-tag**. The last line in the “customer” screen template places the contents of the field starting at location 2,48 (the number 2486.14) into the next output buffer.

The reset-action template is designed to return the session to the “acct” application’s menu screen where it awaits the next transaction. Figure 3-12 shows the reset-action template, and its accompanying screen templates.

Figure 3-12
Reset-action template for accounting application

Action Template

```
#reset-action template for the getbalance
#action template, filename: clr_cust.act
clr_cust  accounting  —  —
clrcust   #exits customer info screen
atacctmenu #validates application menu screen
```

Screen Template

```
#screen template to clear customer info
#screen, filename: clrcust.scn
clrcust  1,1  Account Number:
0,0      —   4
0,0      >   ENTER
```

```
#screen template that validates the acctng
#menu screen, filename: atacctmenu.scn
atacctmenu 1,20 ACME Accounting
```

Notice that the **atacctmenu** screen used in the reset-action template is the same screen used in the initial-action template. This screen template verifies that the menu screen has returned, but it performs no function.

Figure 3-13 shows the logout-action template, and its accompanying screen templates, that return the host computer to the login screen if an error occurs and the reset-action template also experiences an error.

Figure 3-13
Logout-action template for accounting application

Action Template	Screen Template
<pre>#logout-action template for the getbalance #action template, filename: logout_cust.act logout_cust accounting — — clrcust #exits customer info screen clrmenu #exits applicatoin menu logout #logouts out to prepare for login</pre>	<pre>#screen template to clear customer info #screen, filename: clrcust.scn clrcust 1,1 Account Number: 0,0 — 4 0,0 > ENTER</pre>
	<pre>#screen template to exit the #accounting application #filename: clrmenu.scn clrmenu 1,20 ACME Accounting 0,0 — 5 0,0 > ENTER 0,0 — Y 0,0 > ENTER</pre>
	<pre>#screen template to log off the host #computer, preparing for the initial #action template to be executed #filename: logout.scn logout 0,0 — 0,0 — logout 0,0 > ENTER</pre>

The logout-action template brings the host computer back to the login screen. The TRS then executes the initial-action template, as it does whenever any logout-action template is successful.

If the transaction fails before the customer screen is accessed, the logout-action template searches the other screen templates until a valid tag match is found, then it executes that screen template and the following screen templates.